CollA: 다중 프로세싱 기반 연합학습 프레임워크 소개

김동희*, 김광수
성균관대학교
*ym.dhkim@skku.edu, kim.kwangsu@skku.edu

CollA: Introduce of Multi-Processing Based Federated Learning Framework

Kim Dong Hee*, Kim Kwang Su
SungKyunKwan Univ.

요 약

연합학습은 분산된 환경에서 직접적인 데이터 전송 없이 각 클라이언트에서 학습한 가중치만 활용하여 전체 모델 성능을 향상시키는 새로운 학습 방법이다. 연구 목적으로 연합학습을 수행 시에는 물리적 문제와 비용적인 문제로 인하여 사실상 수백, 수천 개의 가상 클라이언트를 구성하는 것은 불가능하다. 따라서, 시스템 자원을 가용 할 수 있는 범위 내에서 가상 클라이언트를 최대한 생성 가능한 프레임워크가 필요하다. 또한, 로컬에서 다수의 클라이언트가 동시에 학습 시, 모델 간에 가중치를 공유하거나, 리소스를 공유 할 때 충돌이 없어야 한다. 마지막으로 연합학습의 새로운 방법론을 쉽게 적용 할 수 있어야한다. 본 논문에서는 Tensorflow와 Ray를 활용하여 자체 개발 한 다중 프로세싱 기반 연합학습 프레임워크를 소개한다.

Ⅰ. 서 론

연합학습(Federated learning)은 분산된 환경에서 데이터의 직접적인 전송 없이 각 클라이언트에서 학습한 가중치만 전송하여 전체 모델 성능을 향상시키는 새로운 학습 패러다임이다 [1]. 이를 통해 각 클라이언트에서 보유하고 있는 데이터를 외부에 공개하지 않아도 되기 때문에 프라이버시 보호에 유리하며, 대용량 데이터를 중앙에 집중하지 않음으로써 네트워크 자원과 스토리지 자원을 낭비하지 않을 수 있다.

연구 목적으로 연합학습을 수행하고자 할 때는 클라이언트가 많은 경우, 데이터 분포가 고르지 않은 경우, 일부 데이터만 보유하고 있는 경우 등 다양한 학습 환경을 고려해야한다 [2]. 그러나 물리적인 문제와 비용적인 문제로 인하여 사실상 수백, 수천 개의 클라이언트를 구성하는 것은 불가능하다. 그러므로 연합학습을 활용한 서비스를 연구 및 개발하는 과정에서 다양한 연합학습 환경을 제공 할 수 있는 프레임워크를 활용해야한다.

기존 연합학습 연구에서 사용하고 있는 프레임워크로는 Tensorflow Fed [3], PySyft [4], Flower [5]가 있다. 이러한 프레임워크는 기 연구된 내용을 활용하여 서비스를 개발하기에 활용성과 기능적인 측면에서 부족함이 없으나, 연합학습 성능 향상을 위해 가중치 취합, 압축, 정규화 등 새로운 방법론을 창출함에 있어 사용자화(customize)가 어렵고, 기능 구성이 복잡하여 초보자는 프레임워크를 숙지하는데 많은 시간을 소요해야한다. 따라서 사용자화가 쉽고, 초보자도 쉽게 활용 할 수 있는 프레임워크 개발이 필요하다.

이처럼 연합학습 프레임워크를 연구에 활용하기 위해서는 1) 다양한 데이터 분포를 갖는 클라이언트를 선언 할 수 있어야하고, 2) 로컬 모델 학습 시 클라이언트 간 충돌이 없어야하며, 3) 새로운 기능을 손쉽게 추가 할 수 있어야 한다. 본 논문에서는 세 가지 조건을 충족하는 다중 프로세싱 기반의 새로운 연합학습 프레임워크 CollA(Collaborative AI)를 소개한다. 자체 개발 한 CollA는 시스템 자원이 허용하는 한, 무한대의 클라이언트를 생성 할 수 있고, 멀티 프로세싱 환경에서 충돌을 완벽히 회피하며, 기존의 Tensorflow API를 그대로 수용함으로써 기능 추가 및 활용이 용이하다.

Ⅱ. Tensorflow의 다중 프로세싱 문제 및 해결방안

가상의 클라이언트들이 각자의 모델을 학습하기 위한 방법으로 대부분 다중 프로세싱을 사용한다. 하지만 PyTorch와 달리, Tensorflow에서 Python의 다중 프로세싱을 사용하면 충돌 문제가 발생한다. Tensorflow는 모델을 학습하는 과정에서 별도의 쓰레드(Thread)를 생성하는데, Python에서 다중 프로세싱으로 모델 학습을 진행하면, python의 child 프로세스가 Tensorflow의 쓰레드가 끝나는 시점을 인식하지 못해서 join 함수를 호출하더라도 끝나지 않고 좀비 프로세스로 남는다 [그림 1-(a)].

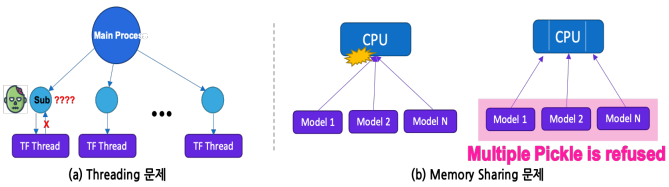

그림 1 Tensorflow Multi-processing의 문제점

더 나아가, Tensorflow는 모델 그래프를 생성하고 이를 미리 저장(pickle)하는데, soft-copy를 하면 pickle된 모델끼리 메모리를 공유하면서 충돌이 발생한다. 이로 인해 학습 가중치를 공유하고, 결국에는 연합학습 효과가 아닌 중앙 집중 학습과 동일한 구성을 이룬다. 반면, Deep-copy를 하면 메모리 공유는 피할 수 있으나, 다중 pickle로 인해 python 인터프리터에서 오류를 일으킨다 [그림1-(b)].

이 문제는 분산 어플리케이션 개발을 위한 오픈소스 프레임워크 Ray [6]를 통해 해결 가능하다. Ray는 인공지능의 분산학습을 위한 전용 프레임워크다. CollA는 Ray를 활용하여 다중 프로세싱 환경을 구성하고, CPU 코어당 하나의 클라이언트를 배정하여 독립적으로 학습을 진행한다.

Ⅲ. CollA (Collaborative AI) 프레임워크

CollA는 그림 2와 같이 세 가지 계층으로 구성한다. 첫 번째 계층은 연합학습의 최적화 기능, 가상 클라이언트 정의, 스케쥴러, 학습 함수 등을 포함하는 Federated Core이고, 다음으로 데이터의 분포, 클래스의 수, 모델 등 학습에 필요한 전처리기를 포함하는 Middle layer가 있다. 마지막으로 서비스 어플리케이션이 구동하는 Application layer로 구성한다.

CollA의 Middle layer 중 Utils 모듈은 공개되어있는 데이터를 다운 받고, 클라이언트에 배정하기 위한 다양한 데이터 분포를 생성한다. 연합학습에서 데이터 분포를 다르게 하는 것은 1) 클라이언트 별 보유 클래스 수를 다르게 하거나 , 2) 클라이언트 별 보유 샘플 수를 다르게 하는 것 (또는 둘 다)을 말한다. 이 때, 로딩 시간과 분배하는 시간을 줄이기 위해 캐쉬 기능을 제공한다. 또한, local 학습에 필요한 공통 설정(optimization, loss, local epochs 등)은 Config 모듈에서 처리한다.

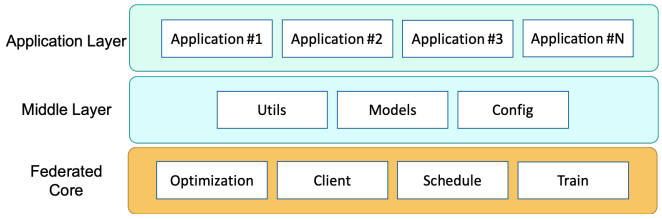

그림 2 CollA Stack

연합학습 프레임워크에서 다뤄야할 문제는 제한된 리소스 환경에서 수많은 클라이언트의 학습을 진행해야하는 점이다. 즉, 가상 클라이언트를 메모리 리소스가 허용하는 범위 내에서 많이 생성 가능해야하며, CPU 리소스의 스케쥴링을 통해 효율적으로 학습을 진행해야한다. CollA는 주요 연합학습 메커니즘을 Federated Core 계층에서 지원한다. 이 모듈은 가상의 클라이언트를 생성하고, CPU 자원을 점유하는 프로세스를 생성하며, 클라이언트가 프로세서를 안전하게 할당받아 모델을 학습 할 수 있도록 하는 스케쥴링 메커니즘을 제공한다.

스케쥴러는 학습을 위한 CPU 코어를 할당 받은 Ray Agent와 데이터와 모델 가중치를 보유하고 있는 클라이언트를 매칭하는 역할을 수행한다 [그림 3]. 스케쥴링 첫 단계로, 클라이언트들은 보유한 모델의 가중치와 데이터를 Ray Agent에 전달한다. 이 후, Ray Agent는 전달 받은 데이터와 모델 가중치를 사용하여 다른 클라이언트와 독립적으로 학습을 진행한다. 학습을 완료하면, Ray Agent는 클라이언트 모델에 학습된 가중치를 설정한다. 그리고, 학습을 아직 진행하지 못한 클라이언트에게 같은 방법으로 자원을 할당한다. 모든 클라이언트가 로컬 학습을 완료하면, 클라이언트들에게 가중치를 요청 받아 평균(FedAvg [7])을 계산하고, 결과를 배포한다. 연합학습에서는 이 과정을 Global round라 하고, 일반적으로 수십, 수백 번 반복하면서 공통 모델의 학습을 진행한다.

CollA는 CPU 코어를 할당받은 프로세스가 다른 모델과 독립적으로 학습함으로써 충돌을 회피한 것이 특징이다. 이를 위해 CollA에서는 모델을 직접 전달하지 않고 가중치를 전달한다. 모델을 직접 전달하는 경우, 참조관계 충돌이 발생하고, Tensor graph를 생성에 상당한 시간이 소요되기 때문이다. 이런 이유로 Ray Agent는 학습을 위한 더미 모델을 한번만 선언하고, 클라이언트로부터 가중치를 전달 받아 학습을 진행한다.

CollA는 데이터 전처리부터 클라이언트 생성, 로컬 및 global 학습까지 모두 기존의 Tensorflow API를 그대로 사용하기 때문에 Tensorflow에 기초 지식이 있는 사용자면 누구나 쉽게 활용 가능하다. 또한, 새로운 방법의 적용과 기존의 여러 모델 및 학습 방법론을 적용이 용이하다.

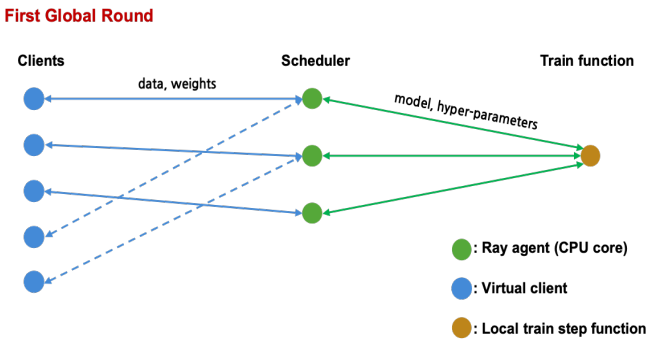

그림 3 CollA 스케쥴러 메카니즘 (1 global 라운드 예시)

Ⅳ. 결론

연합학습은 데이터를 직접 전송하지 않고, 가중치만 전송하여 개인정보 문제, 자원 효율성 문제를 해결하고자 하는 새로운 패러다임이나, 현재 연구 단계에 있는 기술이다. 본 기술을 연구하기 위해서는 가상의 클라이언트를 가능한 많이 생성 할 수 있어야하며, 서로 간의 충돌 없이 학습이 가능해야하고, 새로운 기능을 쉽게 추가 할 수 있어야한다. 본 논문에서는 Ray를 활용한 멀티 프로세싱 기반 연합학습 프레임워크 구조와 스케쥴링 메커니즘을 소개했다. 제안한 프레임워크를 활용하면 리소스가 허용하는 만큼 가상 클라이언트를 생성 할 수 있고, 모델 간 충돌 없이 각 클라이언트 모델 학습이 가능하다. 또한, 기존의 Tensorflow API를 그대로 차용하기 때문에 Tensorflow에 기초가 있는 사용자라면 누구나 쉽게 활용 가능하다.

ACKNOWLEDGMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신산업진흥원의 지원을 받아 수행된 헬스케어 AI 융합 연구개발 사업임 (No.S0316-21-1006)

참 고 문 헌

[1] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19.

[2] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., ... & Roselander, J. (2019). Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046.

[3] TensorFlow Federated. https://www.tensorflow.org/federated, (last accessed at: 2021.05.18)

[4] PySyft. https://github.com/OpenMined/PySyft, (last accessed at: 2021.05.18)

[5] Flower. https://flower.dev, (last accessed at: 2021.05.18.)

[6] Ray, https://ray.io, (last accessed at: 2021.05.18.)

[7] Konečný, J., McMahan, B., & Ramage, D. (2015). Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575.